IDOL Docker Images

Software Version 12.8

IDOL Docker Technical Note

MICRO FOCUS

Document Release Date: February 2021
Software Release Date: February 2021

Legal notices

Copyright notice

Documentation updates

The title page of this document contains the following identifying information:

- Software Version number, which indicates the software version.
- Document Release Date, which changes each time the document is updated.
- Software Release Date, which indicates the release date of this version of the software.

To check for updated documentation, visit https://www.microfocus.com/support-and-services/documentation/.

Support

Visit the MySupport portal to access contact information and details about the products, services, and support that Micro Focus offers.

This portal also provides customer self-solve capabilities. It gives you a fast and efficient way to access interactive technical support tools needed to manage your business. As a valued support customer, you can benefit by using the MySupport portal to:

- Search for knowledge documents of interest
- Access product documentation
- View software vulnerability alerts
- Enter into discussions with other software customers
- Download software patches
- Manage software licenses, downloads, and support contracts
- Submit and track service requests
- Contact customer support
- View information about all services that Support offers

Many areas of the portal require you to sign in. If you need an account, you can create one when prompted to sign in. To learn about the different access levels the portal uses, see the Access Levels descriptions.

Contents

Introduction

This technical note describes the Micro Focus IDOL Docker images, and how to use them to create and use simple IDOL example instances.

Docker is a platform that allows you to run and use *containers*, which are packages that contain pre-installed and configured software. Containers provide convenient building blocks to allow you to create complex systems more easily.

The IDOL Docker containers allow you to quickly set up a simple IDOL installation, without having to manually install or configure the IDOL components.

> **NOTE:** This technical note assumes that you have installed and set up the Docker platform in your environment. It does not provide general information about how to use Docker.
>
> For more information about Docker, refer to the Docker documentation: https://docs.docker.com/.

The Micro Focus IDOL containers each provide a single IDOL component, with a standard configuration. Micro Focus also provides several Docker Compose files to allow you to link these containers together in simple, standard systems. You can use these containers and Compose files as a starting point for building your own containerized IDOL system.

Available IDOL Containers

The following table describes the available Micro FocusIDOL Docker containers. For more information about the IDOL components, refer to the relevant IDOL documentation.

Container Name	Description
content	IDOL Content component
community	IDOL Community component
agentstore	IDOL Agentstore component, configured to work with the IDOL Community component (for example, the community container)
category	IDOL Category component
categorisation-agentstore	IDOL Agentstore component configured to work with the IDOL Category Component and categorization tasks in IDOL NiFi Ingest (for example, the category, nifi-minimal, and nifi-full containers)
view	IDOL View component
dah	Distributed Action Handler
dih	Distributed Index Handler
nifi-minimal	IDOL NiFi Ingest, with only the IDOL NiFi processors (and not the standard Apache NiFi processors)
nifi-full	IDOL NiFi Ingest, with all the standard processors that are included in the Apache distribution of NiFi, in addition to the IDOL NiFi processors.
find	IDOL Find user interface
mediaserver	IDOL Media Server
mediaserver-enus	IDOL Media Server, configured with the speech packages for US English
mediaserver-playlistserver	IDOL Media Server, with a minimal configuration to serve video for MMAP.
mmap_app	IDOL Media Management and Analysis Platform application
siteadmin	IDOL Site Admin user interface
controller	IDOL Controller component
coordinator	IDOL Coordinator component
dataadmin	IDOL Data Admin user interface

Container Name	Description
statsserver	IDOL Statistics Server
qms	IDOL Query Manipulation Server
qms-agentstore	IDOL Agentstore component, configured to work with the IDOL Query Manipulation Server (for example, the qms container)
omnigroupserver	IDOL OmniGroup Server
eductionserver	IDOL Eduction Server
answerserver	IDOL Answer Server
answerbank-agentstore	IDOL Agentstore component, configured to work with Answer Bank systems in IDOL Answer Server
passageextractor-agentstore	IDOL Agentstore component, configured to work with Passage Extractor systems in IDOL Answer Server

IDOL Docker Compose Files

The IDOL Docker Compose files allow you to quickly set up a system that uses multiple containers, connected together to provide an IDOL setup.

There are two sets of Docker Compose files:

- `basic-idol`. Set up a simple IDOL installation for ingest, indexing, and query.
- `data-admin` Set up IDOL Data Admin and the IDOL components that it requires to run.

NOTE: The `basic-idol` and `data-admin` names are the folder names in the provided IDOL Docker Compose package. The examples in this technical note are based on these standard names. If you change the folder names, or modify the `COMPOSE_PROJECT_NAME` environment variable, it affects the container names that Docker produces, and you must adapt the example commands.

Basic IDOL Docker Compose

The basic-idol Docker Compose set up has a standard `docker-compose.yml`, which creates a small IDOL setup. This basic setup includes the following containers:

- `content`. The IDOL Content component, for indexing and query.
- `nifi-minimal`. A basic version of IDOL NiFi Ingest, configured with a File System Connector, to allow you to process and ingest files into your IDOL Content component.
- `categorization-agentstore`. An IDOL Agentstore component configured to work with categorization tasks in IDOL NiFi Ingest.
- `find`. The Find user interface, to allow you to view results from your Content index.
- `community`. The IDOL Community component, which Find uses to manage users.
- `agentstore`. An IDOL Agentstore component configured to store users and agent data for Community.
- `view`. The IDOL View component, which Find uses to display document previews.

When you run this container, you can copy files into an internal volume that the File System Connector sends to IDOL NiFi Ingest for processing, and indexing into the IDOL Content component. You can then log in to Find to send queries to find this data.

By default, this setup does not allow you to access the IDOL component ports directly, and does not enable SSL.

Micro Focus also provides several additional compose files to allow you to extend the basic setup. The following table describes these additional files.

Docker File	Description
`docker-compose.add-docsec.yml`	Adds document security for your IDOL documents. This option includes an OmniGroupServer to retrieve user and group information from an LDAP server.
`docker-compose.bindmount.yml`	Provides a bindmount. This option allows you to copy files to a physical directory for the File System Connector to ingest them, rather than using Docker copy.
`docker-compose.expose-ports.yml`	Exposes the IDOL component ports in the containers, so that you can connect to the components directly.
`docker-compose.ssl.yml`	Enables SSL communications for the basic IDOL components.
`docker-compose.add-mmap.yml`	Adds IDOL Media Server to allow you to process image, audio, and video files in addition to text-based content, and the IDOL MMAP application to allow you to perform more advanced analysis on the audio and video.
`docker-compose.add-mmap.ssl.yml`	Enables SSL communications for MMAP and Media Server.

IDOL Data Admin Docker Compose

The `data-admin` Docker Compose set up has a standard `docker-compose.yml`, which creates the required containers to run IDOL Data Admin.

You can optionally also include `docker-compose.ssl.yml` to run IDOL Data Admin and the IDOL components with SSL communications enabled.

Deploy IDOL Containers on Kubernetes

Kubernetes is a system for automating the deployment, scaling, and management of containerized applications. The IDOL containers repository on github provides files to allow you to run the IDOL containers by using Helm, which is a package manager for Kubernetes.

You can view and find the Kubernetes and Helm chart files at:

https://github.com/microfocus-idol/idol-containers-toolkit/tree/main/helm

For more information, view the README file: https://github.com/microfocus-idol/idol-containers-toolkit/blob/main/helm/README.MD.

Requirements

This section describes the requirements for you to run an the basic IDOL Docker Compose set up.

Before you can use the IDOL Docker containers, you must have:

- an installation of Docker, on Windows or Linux. On Linux you must also install Docker Compose. For details about how to install and run Docker and Docker Compose, refer to the Docker documentation.
- an API key to access the Micro Focus IDOL Docker hub to download the container images. You can obtain this key from Micro Focus Support.
- an IDOL License Server, running on a static machine. The IDOL containers use this License Server to retrieve licenses.

To use the IDOL Docker Compose files, you must also obtain the Docker ZIP from Micro Focus Support, which contains the necessary YAML files. For more information about these files, see IDOL Docker Compose Files, on page 8.

Build IDOL Docker Compose System

This section describes how to build and start your IDOL Docker Compose system.

To use the IDOL Docker Compose options, you must obtain the ZIP package from Micro Focus Support, and extract the file to a directory that Docker can access.

When you have this file available, you take the following steps to build the system:

1. Modify the environment file to specify the License Server IP address, and a HTTP Proxy if required. See Modify the Environment Files, below.
2. (`basic-idol` only) Make any required updates to the YAML files for your particular options. You must modify the YAML when you want to use a bindmount, or document security. See Update YAML Files, on the next page.
3. Run the `docker-compose up` command to build and run the containers. See Run Docker Commands, on page 15.

You can also optionally set up SSL/TLS communications between the IDOL components in your containers. In this case, you must provide the required certificates, or a means to generate them. There are also additional environment variables to set. See Use SSL/TLS Communications, on page 16.

Modify the Environment Files

The IDOL Docker Compose ZIP package contains environment files.

For all IDOL Docker Compose options, you must modify the `.env` environment file to provide the details of your license server.

> **NOTE:** To use SSL, you must also modify the `idol-ssl.env` environment file to configure SSL behavior (see SSL Environment Variable Reference, on page 19).

To modify the standard environment file

1. In your IDOL Docker Compose package directory, open the folder for the setup that you want to run (`basic-idol` or `data-admin`).
2. Open the `.env` environment file in a text editor.
3. Update the `LICENSESERVER_IP` variable to give the IP address of your IDOL License Server.

   > **NOTE:** To work with the standard IDOL Docker images, License Server must run on port 20000.

4. (Optional) Modify the following variables for your setup:

HTTP_PROXY	The URL of a HTTP proxy that Docker must use to access the container files on Docker Hub.
IDOL_SERVER_VERSION	The IDOL Server version of the images that you want to download. You can use one of the following version string formats: • 12.*X* Download the latest version of the 12.X release containers. For example, 12.6 downloads the latest available 12.6 version (which might be 12.6.0 or 12.6.1, and so on). • 12.*X.X* Download the latest version of the 12.X.X release containers. For example, 12.6.0 downloads the latest available 12.6.0 version, where the docker containers have been updated but the IDOL component versions have not changed. • 12.*X.X_DockerBuild* Download an exact version of the IDOL Docker containers.

The other environment variables are for internal use, and you do not need to update them unless instructed by Micro Focus Support.

5. Save and close the environment variables file.

Update YAML Files

> **NOTE:** This section applies only to the `basic-idol` Docker Compose setup.

In most cases, you do not need to modify the YAML files for the IDOL Docker Compose package. However, YAML updates are required when:

- you want to run the IDOL NiFi Ingest container with a bindmount. See Modify the YAML to Use a Bindmount, below.
- you want to use LDAP for Find login or document security. See Modify the YAML to Use Document Security, on the next page.

When you want to use these features, you must update these YAML files before you run the `docker-compose up` command

Modify the YAML to Use a Bindmount

To run the IDOL NiFi Ingest container with a bindmount, you must modify `docker-compose.bindmount.yml` to add the directory to use.

The bindmount means that you have a directory on the computer where you run your Docker containers that you can use to ingest data into the IDOL NiFi Ingest containers. In the default configuration, you upload documents to ingest by using a Docker copy.

To run with a bindmount

1. In your IDOL Docker Compose ZIP package, open the `basic-idol` folder.
2. Open `docker-compose.bindmount.yml` in a text editor.
3. Under `idol-ingest-volume`, update the device: property with the directory that you want to use for ingest. For example:

```
volumes:
   idol-ingest-volume:
      driver_opts:
         type: none
         device: C:\docker\MyIngestDir
         o: bind
         driver: local
```

4. Save and close the YAML file.

Modify the YAML to Use Document Security

To run the IDOL Docker containers with document security, you must modify `docker-compose.add-docsec.yml` to add settings for the LDAP server that you want to use.

The document security setup uses an LDAP server to manage user and group details, and an IDOL OmniGroupServer to expose the users and groups to IDOL.

You can use this option to provide LDAP login for Find. You can also include the IDOL document security options that restrict access to documents. When a user logs into Find, IDOL generates a security string, which it uses whenever the user makes a query, to ensure they can only access permitted documents.

> **NOTE:** By default, the connector configurations in the IDOL Docker container do not include document security. You must configure the connectors after you set up the containers. For more details about how to configure the connectors in IDOL NiFi Ingest, refer to the IDOL NiFi Ingest documentation.

To run with document security

1. In your IDOL Docker Compose ZIP package directory, open the `basic-idol` folder.
2. Open `docker-compose.add-docsec.yml` in a text editor.
3. Under `x-args-security`, update the following parameters for your system:
   - `LDAP_SERVER`. The IP address or host name of your LDAP server.
   - `LDAP_PASSWORD`. The password for the Base DN user in LDAP (that is, the user that can access all documents in your server).

     > **IMPORTANT:** Encrypt this password by using the Micro Focus `Autpassword` command-line tool. This tool can generate the AES encryption keys, and the encrypted password

> strings.
>
> You supply the AES keys to OmniGroupServer by replacing the `basic-idol/omnigroupserver/aes.key` file in your IDOL Docker Compose ZIP package directory. Do not use the `aes.key` file provided; it is a placeholder file only.

```
x-args-security:
  # Put configuration details for the ldap server here
  - &ldap-server "LDAP_SERVER=myldap.example.com"
  ...
  # Don't put the LDAP password here in plain text:
  # encrypt your password via autpassword with omnigroupserver/aes.key
  - &ldap-password "LDAP_PASSWORD=qNvqIYaYxZyOEDrmz/gthg=="
```

4. To provide document security, uncomment (delete the #) for the following parameters:

   ```
   - &document-security-type "DOCUMENT_SECURITY_TYPE=NT_V4"
   - &document-security-type-mode "DOCUMENT_SECURITY_TYPE_MODE=AUTONOMY_SECURITY_V4_NT_MAPPED"
   - &document-security-type-propmatch "DOCUMENT_SECURITY_TYPE_PROPMATCH=nt_v4"
   ```

   Modify the values for your configuration. These parameters are used by the IDOL Content component and IDOL Community component for document security. For more information, refer to the *IDOL Content Component Reference* and *IDOL Community Component Reference*.

Run Docker Commands

After you have updated the environment file with your License Server details, and made any optional changes, you can run the containers.

To run the Docker Compose system

1. Open a command prompt.
2. Log in to docker by typing the following command:

   ```
   docker login -u microfocusidolreadonly
   ```

   When prompted, type the API key provided by Micro Focus Support.

   This log in allows you to access the IDOL containers in the `microfocusidolserver` organization on Docker Hub.
3. In the command prompt, switch to the directory for the Docker Compose system that you want to run (`basic-idol` or `data-admin`).

   > **NOTE:** Docker uses these folder names to create the names of the container instances. If you change the default `basic-idol` or `data-admin` folder names or modify the `COMPOSE_PROJECT_NAME` environment variable, you must adapt some of the examples in later sections of this technical note.

4. Run the `docker-compose up` command to build and run the containers:

```
docker-compose up
```

This option builds the simple system.

To add any of the additional features, you can use the `-f` argument with the name of each YAML file that you want to use. For example, for `basic-idol`, the following command runs the basic system with a bindmount and exposes the component ports:

```
docker-compose -f docker-compose.yml -f docker-compose.expose-ports.yml -f
docker-compose.bindmount.yml up
```

For details of the available YAML files, see IDOL Docker Compose Files, on page 8.

Check that the System is Running

After the up command is complete, you can check that your IDOL Docker system is running correctly by accessing one of the available user interfaces.

- For `basic-idol`: `http://`*`DockerHost`*`:8080/find/`
- For `data-admin`: `http://`*`DockerHost`*`:8080/dataadmin/`

Where *`DockerHost`* is the IP address or host name of the computer where you have run the docker containers.

Use SSL/TLS Communications

IDOL components and front end applications support SSL/TLS.

To configure the docker compose set up to use SSL/TLS

1. Modify the `idol-ssl.env` environment file to configure the SSL behavior.

   The `idol-ssl.env` file is the environment file for all the services that Docker Compose creates. This file configures the environment variable configurations that are common to all services. If required, you can also set environment variables for individual services. See SSL Environment Variable Reference, on page 19.

2. Modify the `docker-compose.ssl.yml` file to define how to set up the deployment with certificates.

   You can provide these certificates (recommended), or you can provide an OpenSSL-based Certificate Authority to generate the certificates when you run the containers. See Set Up SSL/TLS Certificates and Trust Stores, on the next page.

3. Modify the `docker-compose.ssl.yml` file to configure the `ssl-volume` bind volume, which provides the certificates from the host machine to the containers. By default this has the following configuration:

   ```
   x-ssl-volume: &ssl-volume
      type: bind
   ```

```
source: ../ssl/intermediate
target: /ssl
```

You must adjust the `source` path to the appropriate location of certificates for your system.

> **NOTE:** Micro Focus recommends that you leave the `target` path as `/ssl`, which is required for `idol-nifi` to retrieve certificates. In all cases, the `target` path must correspond to the directory that you set in the `IDOL_SSL_CA_MOUNTDIR` environment variable (see SSL Environment Variable Reference, on page 19).

4. Send the docker compose up command, including the `docker-compose.ssl.yml` file. For example:

   ```
   docker-compose -f docker-compose.yml -f docker-compose.ssl.yml up
   ```

   For the `basic-idol` setup, if you also want to include MMAP, you must also add the `docker-compose.add-mmap.ssl.yml` file to the `up` command.

   > **NOTE:** Between the `ssl-volume` and the environment variables that you provide to the containers, the containers must be able to either find or generate a certificate for the component. If it cannot obtain an appropriate certificate, the `docker-compose up` command exits with an error.

Set Up SSL/TLS Certificates and Trust Stores

When you want to run the IDOL Docker Compose packages with SSL, you must either provide the SSL certificates (recommended), or provide an OpenSSL-based Certificate Authority to generate the certificates when you run the containers.

> **NOTE:** If you configure the containers to generate the certificates, you must provide a certificate for your certificate authority that is able to sign Certificate Signing Requests.

You use the `idol-ssl.env` environment variables file to provide information about your certificates. The `docker-compose.ssl.yml` defines an `ssl-volume` bind mount, which provides the certificates from the host machine to the containers.

Between `idol-ssl.env` and the `ssl-volume`, the containers must be able to either find a certificate provided for the component, or use the Certificate Authority certificate and configuration to generate one. If the containers cannot obtain an appropriate certificate by either method, the `docker-compose up` command exits with an error.

When you use a custom Certificate Authority to generate your certificates, you must also set up a trust store for your user interfaces, to allow them to securely communicate with the SSL-activated IDOL components. See User Interface Trust Stores, on the next page.

The following sections describe how to provide or generate the certificates, and trust stores.

Provide Certificates

Micro Focus recommends that you provide certificates for the container.

When you start a container with SSL activated, the container looks for a certificate at a configurable path for the component. You configure this path by using the environment variables.

The expected location is:

`${IDOL_SSL_CA_MOUNTDIR}/${USER_SSL_CERTS_DIR}/${IDOL_SSL_COMMON_NAME}.cert.pem`

where:

- `${IDOL_SSL_COMMON_NAME}` is the common name for the service.
- `${IDOL_SSL_CA_MOUNTDIR}` and `${USER_SSL_CERTS_DIR}` are environment variable you can use to specify the location of the certificates.

For more information about these environment variables, see SSL Environment Variable Reference, on the next page.

Generate Certificates

When you do not provide a certificate, Docker generates them for the component when you start up the container for the first time. In this case, you must have a Certificate Authority that is able to sign Certificate Signing Requests.

You control the location of the Certificate Authority certificate, and the password for it, by using environment variables.

- The certificate is expected to be in the directory `${IDOL_SSL_SSL_CA_MOUNTDIR}/${IDOL_SSL_CACERT}`.
- The private key file is expected to be in the directory `${IDOL_SSL_SSL_CA_MOUNTDIR}/${IDOL_SSL_CAKEY}`.
- The password for the private key is expected to be set in the environment variable `${IDOL_SSL_CAPASS}`.

User Interface Trust Stores

When you use a custom Certificate Authority to generate certificates, you must set up trust stores for your user interfaces (Find, MMAP, and Data Admin), to allow them to securely communicate with the SSL-activated IDOL components.

You construct the trust store from a PKCS 12 file of the issuing certificate and its private key, along with the root certificate.

To construct the trust store

1. Create a `cfg` directory in the `${IDOL_SSL_CA_MOUNTDIR}` directory. The SSL generation scripts require this directory to store the certificate generation configuration.
2. Generate an intermediate PKCS 12 file. The following example command creates a file, `intermediate.pkcs12`:

   ```
   openssl pkcs12 -export -in certs/intermediate.cert.pem -inkey
   private/intermediate.key.pem -passin pass:PRIVATE_KEY_PASSWORD -out
   certs/intermediate.pkcs12 -passout pass:PASSWORD_FOR_PKCS12_FILE
   ```

   where *PRIVATE_KEY_PASSWORD* is the password for your private key, and *PASSWORD_FOR_PKCS12_FILE* is your password for the PKCS12 file.
3. Copy your root certificate (for example `ca.cert.pem`) to the intermediate `certs` directory to make

it available to the trust store.

4. Generate the chain file by concatenating the root certificate PEM and the issuing intermediate certificate PEM. For example:

   ```
   cat certs/ca.cert.pem certs/intermediate.cert.pem > ca-chain.cert.pem
   ```

5. Choose the value of the `IDOL_SSL_SUBJ_ALT_NAME` environment variable.

   For a user interface (that is, something that you expect to access from a Web browser), use a wildcard value to generate a wildcard certificate. For example, if the computer hosting the certificate is accessible on the exampl.com domain, use `*example.com`. This option prevents warnings from your browser because of a difference between the domains that the certificate is valid for, compared with the domains that it is accessible on.

   For a back end component, the value that you choose is less important, because for this demo system the communications between the back end components and the user interfaces are not verified against the server they run on.

6. Import your chain certificate into your browser. This step prevents warnings about the issued certificates being untrusted.

SSL Environment Variable Reference

The following table lists the environment variables that are required for SSL/TLS communications to work in your docker environment.

To configure these values for all services, you set the environment variable in the `idol-ssl.env` file.

If required, you can also modify the environment variables for an individual service by creating an `environment` section in the `docker-compose.ssl.yml` file section for that service. However, in most cases the default values are suitable and this approach is not required.

> **IMPORTANT:** The `idol-nifi` container has its own environment variable, `USE_SSL` to activate SSL, and it uses the `ssl-volume` bind mount to retrieve certificates. NiFi Ingest does not use the other environment variables listed here.
>
> NiFi Ingest uses SSL only to communicate with your IDOL components, and does not use HTTPS to restrict access to the user interface.

Variable	Description
`IDOL_SSL`	You must set this value to activate SSL.
`IDOL_SSL_SUBJ_ALT_NAME`	The second DNS Name entry in Subject Alternative Name in the certificate. This value is service-specific.
`IDOL_SSL_CACERT`	The directory path to the issuing certificate for the CA, relative to `IDOL_SSL_SSL_CA_MOUNTDIR`. This value is specific to your CA setup.
`IDOL_SSL_CAKEY`	The directory path to the private key for the issuing certificate for the CA, relative to the `IDOL_SSL_SSL_CA_MOUNTDIR`. This value is specific to your CA setup.

Variable	Description
IDOL_SSL_CAPASS	The password for IDOL_SSL_CAKEY. This value is specific to your CA setup.

The following table describes optional environment variables, which you might need to change to match any differences for your Certificate Authority setup.

Variable	Default	Description
IDOL_SSL_CA_MOUNTDIR	/ssl	The directory where the bind mount for the SSL Certificate Authority is mounted on the container. This value is container-specific.
USER_SSL_CERTS_DIR	certs	The directory path where the containers can generate and find certificates, relative to IDOL_SSL_CA_MOUNTDIR. This value is specific to your CA setup.
IDOL_SSL_COMMON_NAME	idol-${IDOL_COMPONENT}	The COMMON_NAME for the certificate. Docker populates the value of ${IDOL_COMPONENT} internally when it builds the component container. This value is also the first DNS Name entry in Subject Alternative Name, for example idol-content.
USER_SSL_CRL_DIR	crl	The directory path to use to store CRL information, relative to IDOL_SSL_CA_MOUNTDIR. This value is specific to your CA setup.
USER_SSL_NEWCERTS_DIR	newcerts	The directory path to use to generate newcerts, relative to IDOL_SSL_CA_MOUNTDIR. This value is less important than USER_SSL_CERTS_DIR, and is specific to your CA setup.
USER_SSL_DATABASE_FILE	index.txt	The file name of the database of issued certificates for your Certificate Authority. This value is specific to your CA setup.
USER_SSL_SERIAL_FILE	serial	The file name of the serial file for your Certificate Authority. This value is specific to your CA setup.
USER_SSL_RANDFILE	private/.rand	The directory path to the randfile for your Certificate Authority, relative to IDOL_SSL_CA_MOUNTDIR. This value is specific to your CA setup.
USER_SSL_CRLNUMBER_FILE	crlnumber	The file name of the CRLNumber file for your Certificate Authority. This value is specific to your CA setup.
USER_SSL_CRL_FILE	crl/intermediate.crl.pem	The directory path to the CRL file for your Certificate

Variable	Default	Description
		Authority, relative to `IDOL_SSL_CA_MOUNTDIR`. This value is specific to your CA setup.
`USER_ISSUING_CA_PKCS12`	`intermediate.pkcs12`	The directory path to the PKCS 12 file of the issuing certificate and private key, relative to `IDOL_SSL_CA_MOUNTDIR/USER_SSL_CERTS_DIR`. This value is specific to your CA setup.
`ROOT_CERTIFICATE`	`ca.cert.pem`	The directory path to a copy of the root certificate PEM file, relative to `IDOL_SSL_CA_MOUNTDIR/USER_SSL_CERTS_DIR`. This value is specific to your CA setup.
`CA_CHAIN_FILE`	`ca-chain.cert.pem`	The directory path to a chain PEM file for the issuing and root certificates, relative to `IDOL_SSL_CA_MOUNTDIR/USER_SSL_CERTS_DIR`.

Use the Basic IDOL Docker Setup

This section describes how to use the `basic-idol` Docker Compose setup to index and search files, and how to access the Find and IDOL NiFi Ingest user interfaces.

Index Documents

After you have built your IDOL Docker Compose system, you can index documents.

Index Documents with Docker Copy

In the default IDOL Docker Compose setup, IDOL NiFi Ingest has an internal volume that you can use to index documents. You must transfer documents to this directory by using the docker cp command.

To index a document by using docker copy

1. Open a command prompt.
2. Run the docker cp command to transfer a file to the `basic-idol_idol-nifi_1:/idol-ingest/` volume. For example:

   ```
   docker cp example.pdf basic-idol_idol-nifi_1:/idol-ingest/
   ```

   Where your setup include Media Server, you can use the same process to ingest media files. For example:

   ```
   docker cp example.mp4 basic-idol_idol-nifi_1:/idol-ingest/
   ```

After you have run the copy, IDOL NiFi Ingest retrieves the documents from this directory and ingests them into your system.

Index Documents with a Bindmount

When you use a bindmount in your IDOL Docker Compose setup, you can copy files directly to a directory on the computer. IDOL NiFi Ingest retrieves them from this directory and ingests them into your system.

Access User Interfaces

The IDOL user interfaces available in the `basic-idol` IDOL Docker Compose system allow you to view your ingest stream, and search your documents. Where your setup includes Media Server and MMAP, you can also use the MMAP video player to view the indexed video.

Use IDOL NiFi Ingest

IDOL NiFi Ingest is a user interface that allows you to set up an ingest process. It provides connector processors that can retrieve documents from many different repositories, extract the useful content, and index it into IDOL.

In the IDOL Docker Compose system, you can access it at the following URL:

`http://DockerHost:8080/nifi/`

Where *DockerHost* is the IP address or host name of the computer where you have run the docker containers.

The IDOL NiFi Ingest user interface allows you to view and modify the configured ingestion chain, and monitor the process.

In the basic system, the default ingest process indexes documents from the internal ingest volume or bindmount. It also performs Eduction to extract personally identifiable information (PII) from the documents that you index.

When you use Media Server as well, it has additional processes to detect media files, and route them to Media Server for analysis, including Speech to Text for video files.

For information about how to modify the ingestion process in the NiFi setup, refer to the IDOL NiFi Ingest documentation.

Use IDOL Find

IDOL Find is a search user interface that allows you to search and view documents. It also provides some graphics and text analytics, such as topic maps and sunburst diagrams to provide more insight into your search results.

In the IDOL Docker Compose system, you can access it at the following URL:

`http://DockerHost:8080/find/`

Where *DockerHost* is the IP address or host name of the computer where you have run the docker containers.

In the default setup, you can log in to Find by using the default login details, username **admin** and password **admin**.

> **NOTE:** Find uses the IDOL Community component to store user details. For a more advanced system with multiple users, you can use the document security setup to add an LDAP server. In this case, you can log in to Find by using the user and password for a valid LDAP user. In this case, all users have the basic Find role, rather than the admin role, so the interface has different available features.
>
> You can also add users to Community directly. In this case, you must run the `docker-compose up` command with the component ports exposed, by using the `docker-compose.expose-ports.yml` option. You can then manually add users to Community.

After you log in you can search for documents and view the graphs and options. When you use Media Server, Find can return your media files as well, and uses the MMAP video player to provide more advanced video playback and search options.

Use the Data Admin Docker Setup

This section describes how to use the `data-admin` Docker Compose setup.

The `data-admin` setup provides all the IDOL components required to run IDOL Data Admin. However, the setup does not provide a built-in way to add any additional components or information (for example, it does not provide the ability to index documents into Passage Extractor).

Access the IDOL Data Admin User Interface

The IDOL Data Admin user interface allows you to access and use IDOL search optimization tools such as promotions, and to use IDOL Answer Bank to create and maintain a set of questions and answers for natural language question answering.

In the IDOL Docker Compose system, you can access it at the following URL:

`http://`*`DockerHost`*`:8080/dataadmin/`

Where *`DockerHost`* is the IP address or host name of the computer where you have run the docker containers.

You can log in to IDOL Data Admin by using the default login details, username **admin** and password **admin**.

Troubleshoot Common Problems

This section provides some information about some common problems that might occur when you build and run your Docker Compose setup.

On Windows, IDOL NiFi Ingest or MMAP volumes are not created

During the up command, you might get Windows prompts to allow file sharing access to particular directories. Access to these directories is required for the volumes that IDOL NiFi Ingest and MMAP use.

If you have this issue, re-run the `up` command and ensure that you click these file sharing prompts to allow access.

Applications are very slow or unresponsive

You might need to increase the amount of memory available to your applications, by using the Docker settings.

Customize the IDOL Docker Setup

The IDOL Docker Compose files provide an example system to demonstrate one way to run IDOL in containers. In most cases, to run a containerized IDOL in a production environment you must customize the setup.

The scope of this document does not extend to the kind of advanced customizations that you might want to make.

For general information about docker containers, refer to the Docker documentation.

If you require further assistance modifying your IDOL Docker container environments, contact Micro Focus Support.

Send documentation feedback

If you have comments about this document, you can contact the documentation team by email. If an email client is configured on this system, click the link above and an email window opens with the following information in the subject line:

Feedback on Micro Focus IDOL Docker Images 12.8 IDOL Docker Technical Note

Add your feedback to the email and click **Send**.

If no email client is available, copy the information above to a new message in a web mail client, and send your feedback to swpdl.idoldocsfeedback@microfocus.com.

We appreciate your feedback!